Jyväskylän yliopiston julkaisuarkisto
Jyväskylä University Digital Archive

UNIVERSITY OF JYVÄSKYLÄ

This is an electronic reprint of the original article.
This reprint *may differ* from the original in pagination and typographic detail.

Author(s): Juvonen, Antti; Hämäläinen, Timo

Title: An Efficient Network Log Anomaly Detection System using Random Projection Dimensionality Reduction

Year: 2014

Version:

Please cite the original version:

Juvonen, A., & Hämäläinen, T. (2014). An Efficient Network Log Anomaly Detection System using Random Projection Dimensionality Reduction. In M. Badra, & O. Alfandi (Eds.), 2014 6th International Conference on New Technologies, Mobility and Security (NTMS) : Proceedings of NTMS'2014 Conference and Workshops. IEEE. https://doi.org/10.1109/NTMS.2014.6814006

All material supplied via JYX is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

An Efficient Network Log Anomaly Detection System using Random Projection Dimensionality Reduction

Antti Juvonen, Timo Hämäläinen
Department of Mathematical Information Technology
University of Jyväskylä
FI-40014 Jyväskylä, Finland
Email: {antti.k.a.juvonen, timo.hamalainen}@jyu.fi

***Abstract*—Network traffic is increasing all the time and network services are becoming more complex and vulnerable. To protect these networks, intrusion detection systems are used. Signature-based intrusion detection cannot find previously unknown attacks, which is why anomaly detection is needed. However, many new systems are slow and complicated. We propose a log anomaly detection framework which aims to facilitate quick anomaly detection and also provide visualizations of the network traffic structure. The system preprocesses network logs into a numerical data matrix, reduces the dimensionality of this matrix using random projection and uses Mahalanobis distance to find outliers and calculate an anomaly score for each data point. Log lines that are too different are flagged as anomalies. The system is tested with real-world network data, and actual intrusion attempts are found. In addition, visualizations are created to represent the structure of the network data. We also perform computational time evaluation to ensure the performance is feasible. The system is fast, finds real intrusion attempts and does not need clean training data.**

***Keywords*—Intrusion detection, data mining, machine learning, random projection, mahalanobis distance.**

I. Introduction

Web services have become more and more complicated and the amount of network traffic is increasing all the time. This makes ensuring good information security a challenge. In order to detect network attacks and improve security, *intrusion detection systems* (IDS) are used. These systems can generally be divided into two distinct categories: *signature-based* and *anomaly-based* systems [1].

Signature-based intrusion detection is still most commonly used. It uses predetermined attack rules to detect intrusive behavior. Network traffic or other actions are compared to these rules, and if there is a match an alarm is created. The benefits of this approach include fast operation and being able to distinguish different types of attacks based on the rules used to detect them. In addition, the number of false alarms is usually low. However, the attack signatures must be manually created. This means that the signatures can be one step behind attackers, and new unknown vulnerabilities can be exploited until a suitable rule is generated and the IDS rule set updated. Anomaly-based systems are based on a different principle. New incoming traffic or behavior is compared to the normal profile, and if an action deviates from the norm, it is flagged as an anomaly. Consequently, new and previously unknown intrusion attempts can be detected. The network profile can be updated periodically or in real-time, which means that the system adapts to changes in network traffic. On the other hand, some algorithms used in anomaly detection systems can be too slow for real-time detection. In addition, the number of false alarms can be unpractically high if the system is not configured properly. Many possible algorithms and methods can be found in Section II. It also possible to combine different detection principles into a hybrid IDS (HIDS) [2].

One big issue with anomaly detection systems is the efficiency and speed. If the amount of network traffic is high, it might be impossible to use complicated algorithms fast enough to detect intrusions before it's too late. Many advanced algorithms achieve a high detection rate but are too computationally complex for practical use. In addition, some intrusion detection frameworks can only do batch-analysis of the whole data or require labeled training data.

We propose an anomaly detection framework that deals with these problems. The system preprocesses web server log data and extracts numerical features from it, forming a feature matrix. Then, the dimensionality of the data is reduced using random projection methodology, and a visualization is also obtained to provide information to the network administrator. Subsequently, Mahalanobis distance is used to calculate an anomaly score for each data point. The data points (corresponding to log lines) that have a score higher than set threshold will be flagged as anomalies. The system is very fast and can function even in real-time. When new log lines are introduced, they can be visualized and the anomaly score calculated without starting the analysis from scratch, meaning that new data can be added dynamically. New data points can be added and older ones dropped from the whole dataset, so that the system adapts to changing network traffic over time.

II. Related research

Dimensionality reduction has been widely researched in the intrusion detection context. Perhaps the most well-known method is principal component analysis (PCA) [3], [4], [5]. It has been researched extensively in network anomaly detection [6], [7]. However, it has some problems, such as the fact that it cannot handle nonlinear data. It is also not as fast as random projection.

© 2014 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

This is the authors' postprint version of the article. The original print version appeared as: A. Juvonen and T. Hämäläinen, "An Efficient Network Log Anomaly Detection System using Random Projection Dimensionality Reduction," in *New Technologies, Mobility and Security (NTMS), 2014 6th International Conference on*. IEEE 2014. Available online at http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6814006&isnumber=6813963

Surveys describing advances in the field of intrusion detection have been published [8], [9]. Many machine learning methods, such as self-organizing maps [10] and support vector machines [11] have been used to cluster data and detect anomalies in these systems. Various hybrid systems combining signature and anomaly-based detection have been used [2], [12]. A two-stage adaptive hybrid system for IP level intrusion detection has also been recently devised. A probabilistic classifier detects anomalies and a hidden Markov model narrows down attacker addresses [13]. Recently, genetic algorithms have been widely used in anomaly detection and misuse detection [14], [15]. Another recent development is using artificial immune systems (AIS) in intrusion detection [16].

The authors have already been involved in developing several network anomaly detection systems [17], [18], [19]. These papers mainly focus on diffusion map (DM) methodology for dimensionality reduction. The diffusion map serves the same purpose as random projection in this paper, and DM can be very efficient in finding anomalies and handling outliers in the data. In addition, nonlinear data is not a problem for DM. However, it's main problem is computational complexity, and this limits it's use in real-time anomaly detection. This paper focuses on random projection because of time constraints when analyzing large amounts of traffic. Random projection has not been extensively used in anomaly detection before.

Much of the research related to intrusion and anomaly detection use publicly available datasets, such as DARPA 1998 and DARPA 1999 [20] as well as KDD Cup 99 [21]. However, these datasets have many problems [22], [23] and therefore do not represent real network traffic accurately. We focus on real-world data collected from an actual network.

III. Methodology

In this section, the overall system framework and used methods are explained. Visualization of the whole system can be seen in Figure 1. The system consists of following phases:

- Data acquisition
- Preprocessing
- Feature extraction
- Random projection dimensionality reduction
- Mahalanobis distance score calculation
- Anomaly alerts based on threshold value

First the data must be collected from a network. In this study, Apache HTTP server access logs are used. Data format, preprocessing and feature extraction can vary depending on the dataset.

After acquiring and preprocessing the data, as well as extracting numerical feature matrix from the log files, the dimensionality of the matrix is reduced using random projection. Subsequently, the Mahalanobis distance for each data point from the whole dataset can be calculated. Finally, the data points with Mahalanobis distance higher than a specified threshold value are flagged as anomalies and can be inspected by the network administrator.

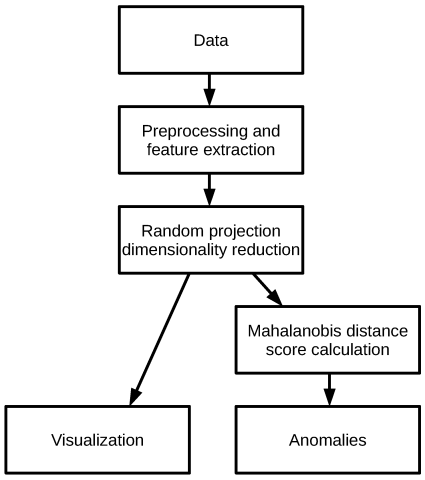

Fig. 1. Overall system framework.

Used data, methods and algorithms are described in more detail in the following subsections.

A. Data acquisition and preprocessing

We use the same network log database that has been used in our previous research [19]. The data comes from a real-life company web server. Different kinds of intrusion attempts and other abnormal log lines are included in the data. We examine a log file that is created in a web server using Apache server software. A single log line uses the following format:

```
127.0.0.1 - -
[01/January/2012:00:00:01 +0300]
"GET /resource.php?parameter1=value1
&parameter2=value2
HTTP/1.1"
200 2680
"http://www.address.com/webpage.html"
"Mozilla/5.0
(SymbianOS/9.2;...)"
```

This format is called Combined Log Format [24]. The used data is from the access logs of the server. These logs contain various information about the network traffic, such as timestamp, HTTP request and the amount of transferred bytes. The logs may contain different intrusion attempts, such as SQL injections, especially in the HTTP request part. For this analysis, the HTTP request string was analyzed and used.

After acquiring the logs, the data is ready for feature extraction. For this analysis, the character distribution is used. This simply means calculating the frequencies of individual characters in the data. These frequencies will form a feature matrix that can be used in the other steps of the analysis. Each row of the column corresponds to an individual log line, and each column corresponds to in individual character. Empty columns corresponding to characters not appearing in the dataset are omitted. This way we get a feature matrix X containing feature vectors $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)$, where each symbol of a vector corresponds to a log line character frequency of one single character. There are d unique characters in the dataset, forming a d-dimensional feature matrix.

B. Random Projection

In random projection (RP), the goal is to project high-dimensional data into a lower-dimensional space using a random matrix [25]. The idea is based on Johnson-Lindenstrauss lemma [26]. It states that points can be projected to a randomly generated subspace and still the distances between points are approximately preserved.

Given the original data with d dimensions, the new subspace has k dimensions so that $k << d$. If the original data matrix is $X_{d\times N}$ and the randomly generated matrix is $R_{k\times d}$, the random projection of the data can be calculated using the following equation [25].

$$X^{RP}_{k\times N} = R_{k\times d} X_{d\times N}$$

As can be seen from the equation, the random projection method is computationally not very expensive even if the original data have a high number of dimensions. However, the generation and orthogonalization of the random matrix R can be complicated, but is not a problem in this case as explained below.

The most important phase of the method is the actual creation of the random matrix R. Basically, R should be orthogonal but unfortunately orthogonalization is computationally expensive. However, a useful result has been presented by Hecht-Nielsen [27]: *"There exists a much larger number of almost orthogonal than orthogonal directions in a high-dimensional space"*. Based on this result, we can assume that orthogonalization can be left out. The practical experimental results done in this paper also support this.

Instead of using Gaussian distributed variables, a much simpler probability distribution has been proposed by Achlioptas [28]:

$$r_{ij} = \sqrt{3} \times \begin{cases} +1 & \text{with probability} \frac{1}{6} \\ 0 & \text{with probability} \frac{2}{3} \\ -1 & \text{with probability} \frac{1}{6} \end{cases}$$

Computing the random matrix with this distribution is very efficient and easy to implement. It is possible to use random projection that is even more sparse. More generally speaking, the items in the random matrix can be calculated using the following probability distribution [29] :

$$r_{ij} = \sqrt{s} \times \begin{cases} +1 & \text{with probability} \frac{1}{2s} \\ 0 & \text{with probability} 1 - \frac{1}{s} \\ -1 & \text{with probability} \frac{1}{2s} \end{cases}$$

It is possible to choose s so that $s \gg 3$. This leads to *very sparse random projections* [29]. However, for this study we use $s = 3$, as proposed by Achlioptas [28].

C. Mahalanobis distance

The Mahalanobis distance [30] is a distance metric that is used for outlier detection in the proposed system. This distance metric takes into account the correlations of the data. The Mahalanobis distance metric is calculated for each individual data point, taking account the distance from the whole dataset. This creates basically an anomaly score. Setting a threshold for this score makes it possible to flag certain data points as anomalies.

If we have a data set X with an individual data vector being $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)^T$, as well as mean $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_N)^T$ and covariance matrix S, the Mahalanobis distance score for each data point can be formally defined as follows [31]:

$$D_M = \sqrt{(\boldsymbol{x} - \boldsymbol{\mu})^T S^{-1} (\boldsymbol{x} - \boldsymbol{\mu})}$$

The outlier detection mechanism used in this paper can be changed, just like other components in the modular system. How the Mahalanobis distance works in practice in this system is described in Section IV.

D. Adding new data points

The methodology described previously is first performed on the whole available data set. However, when new traffic occurs in the network and therefore new data points are generated, the whole analysis does not have to be run from scratch. Preprocessing and character distribution are both trivial for the new log line. Random projection is performed with simple matrix multiplication with the same random matrix created previously. This way the new data point is projected into the same low-dimensional subspace as previous points. Finally, Mahalanobis distance is calculated just like for all the other points.

IV. Experimental results

Apache web server log data was acquired from a real-world company network and preprocessed as explained previously. The test data set that was received contains 1,244,025 lines, and the timespan is about one week. After preprocessing we find that 185 unique characters appear in the data, corresponding to 185 dimensions in the feature matrix.

The data are projected into a 2-dimensional subspace using random projection. Subsequently, Mahalanobis distance is calculated for each data point to form the anomaly score. These distances along with the chosen threshold value can be seen from Figure 2. The values are scaled between 0 and 1 in this figure. Some of the data points seem to be highly anomalous, while most of the points form a large normal cluster. Setting the threshold value higher will mean that only the most anomalous behavior is detected, setting it lower will mean that potentially more anomalies are found but the false alarm rate might increase as well.

Figure 3 shows the 2-dimensional RP visualization, with anomalies highlighted with red. Normal traffic is seen as a big cluster of points, and many queries are far away from the normal cluster, indicating that they are highly anomalous. Using the given threshold value, 278 log lines are flagged as anomalous, meaning that only 0.02% of the traffic is flagged. The other points (99.98%) represent normal traffic.

Because the used dataset is real network data, any prior information about possible intrusions is not available. This is

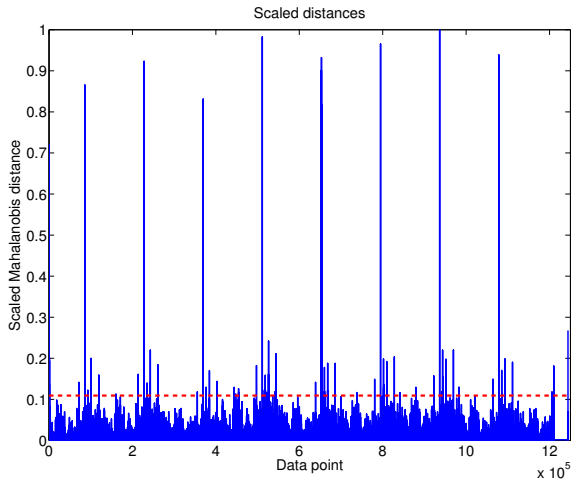

Fig. 2. Scaled Mahalanobis distances with threshold line visible.

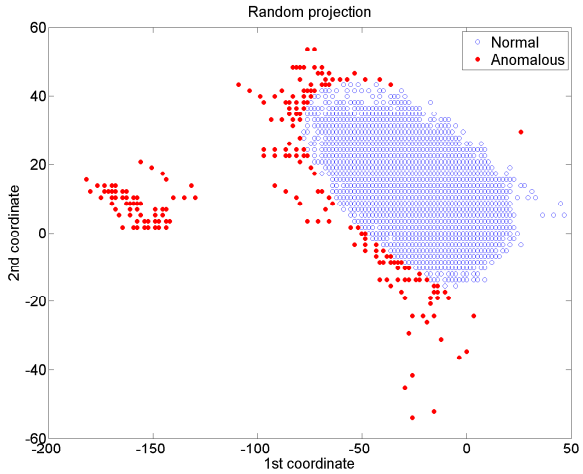

Fig. 3. Random projection visualization of the dataset.

the case in practical situations without artificially generated training data. The 278 anomalies are manually inspected to check if something intrusive is found. Upon inspection it is revealed that only 8 of these loglines are normal and non-intrusive, which equals 2.8% of all the alarms. This suggests that lowering the threshold might reveal more anomalies, even though they are potentially similar to the ones that were already found. The anomalies include GoogleBot scans, as well as several security scans using Nmap, DirBuster and Brutus AET password cracker. The scans mainly focus on finding vulnerabilities in phpMyAdmin software. These intrusion attempts are not very severe for updated systems, and therefore they do not pose a risk at this time. Still, these loglines deviate from normal traffic in a clear way. Any similar scan attempt should be easy to find using the proposed system.

Analyzing the whole data set is very fast. The Python implementation of preprocessing for the whole dataset takes the most time (minutes), while subsequent analysis phases done in Matlab are completed almost instantly. As a comparison to widely used PCA (mentioned in Section II), we calculated

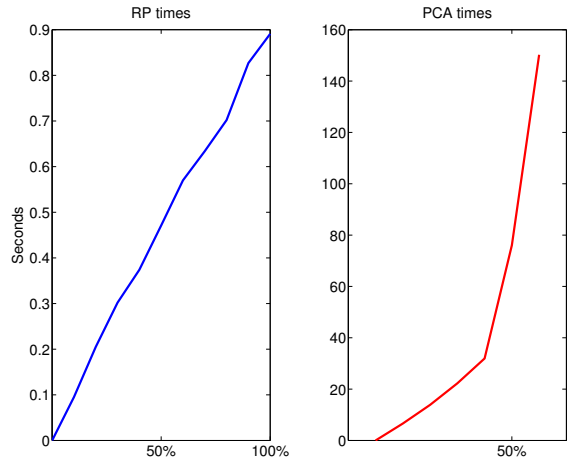

Fig. 4. Computational times for RP and PCA.

some computing times for different sized subsets of the data. This comparison can be seen in Figure 4. The purpose was to compare the time taken for dimensionality reduction, as well as the effect of data size in computational times. It is apparent that RP performs much better, the whole analysis taking less than a second. In addition, the time increase appears to be quite linear, meaning RP has good scalability. PCA analysis is performed only up to 60% of the data set, because the analysis time increases rapidly and becomes unpractically long before even analyzing 100% of the dataset. All of the runs were performed 5 times, and the times were averaged over these 5 runs. It must be noted that Mahalanobis distance calculation is not included in these performance evaluations, because the time taken would have been the same for both RP and PCA matrices.

V. Conclusion

Overall security in a network could be enhanced by using anomaly detection together with traditional signature-based intrusion detection systems. However, anomaly detection systems often use complicated and slow algorithms, which ensures high detection rate but impractically low speed. We propose a framework that can be used to analyze and visualize logs quickly, as well as find anomalous network traffic. This system is not designed to work as the only security measure, but rather as an addition to existing systems.

The system's main advantage is the simplicity and speed. It can easily analyze huge log files with relatively low-powered hardware. In addition, when new network traffic occurs, there is no need to perform the entire analysis from scratch. New data points can be added dynamically and old data can be dropped, so that the system adapts automatically to changing network profile. Also, clean traffic data (traffic that does not contain intrusions) are not needed. However, even though the system was able to generate value by finding intrusion attempts from actual real-world log files, more experiments with new data are needed to ensure that the detection rate is acceptable.

For future research, any component of the framework can be changed. Therefore, different dimensionality reduction and

outlier detection methods could be used. In addition, to make the system more general and avoid overfitting, random projection and subsequent anomaly detection could be performed several times for the same data. After this, each data point would be either flagged as normal or anomalous several times by the system. This is because random projection by definition has a certain random element to it, and might sometimes give unwanted results for individual data points. If the point is flagged as anomalous more times than normal, it will be treated as an anomaly. Bootstrapping is another technique that could be combined with this, making the system even less prone to overfitting. This way, if one random matrix gives unwanted results, it does not lessen the performance of the whole system. These features would make the system more automatic and therefore easier to use for network administrators who are not data mining experts.

ACKNOWLEDGMENT

This research was partially supported by the Nokia Foundation. Thanks are extended to Kilosoft Oy.

REFERENCES

[1] K. Scarfone and P. Mell, "Guide to intrusion detection and prevention systems (idps)," *NIST Special Publication*, vol. 800, no. 2007, p. 94, 2007.

[2] M. A. Aydın, A. H. Zaim, and K. G. Ceylan, "A hybrid intrusion detection system design for computer network security," *Computers & Electrical Engineering*, vol. 35, no. 3, pp. 517–526, 2009.

[3] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[4] H. Hotelling, "Analysis of a complex of statistical variables into principal components." *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933.

[5] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[6] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of PCA for traffic anomaly detection," *ACM SIGMETRICS Performance Evaluation Review*, vol. 35, no. 1, pp. 109–120, 2007.

[7] C. Callegari, L. Gazzarrini, S. Giordano, M. Pagano, and T. Pepe, "A novel PCA-based network anomaly detection," in *Communications (ICC), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–5.

[8] A. Lazarevic, V. Kumar, and J. Srivastava, "Intrusion detection: A survey," *Managing Cyber Threats*, pp. 19–78, 2005.

[9] F. Sabahi and A. Movaghar, "Intrusion detection: A survey," in *Systems and Networks Communications, 2008. ICSNC'08. 3rd International Conference on*. IEEE, 2008, pp. 23–26.

[10] M. Ramadas, S. Ostermann, and B. Tjaden, "Detecting anomalous network traffic with self-organizing maps," in *Recent Advances in Intrusion Detection*, G. Vigna, E. Jonsson, and C. Kruegel, Eds. Springer, 2003, pp. 36–54.

[11] Q. Tran, H. Duan, and X. Li, "One-class support vector machine for anomaly network traffic detection," *China Education and Research Network (CERNET), Tsinghua University, Main Building*, vol. 310, 2004.

[12] H. Om and A. Kundu, "A hybrid system for reducing the false alarm rate of anomaly intrusion detection system," in *Recent Advances in Information Technology (RAIT), 2012 1st International Conference on*, march 2012, pp. 131–136.

[13] R. Rangadurai Karthick, V. Hattiwale, and B. Ravindran, "Adaptive network intrusion detection system using a hybrid approach," in *Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on*, jan. 2012, pp. 1–7.

[14] L. Li, G. Zhang, J. Nie, Y. Niu, and A. Yao, "The application of genetic algorithm to intrusion detection in mp2p network," in *Advances in Swarm Intelligence*, ser. Lecture Notes in Computer Science, Y. Tan, Y. Shi, and Z. Ji, Eds. Springer Berlin Heidelberg, 2012, vol. 7331, pp. 390–397.

[15] M. Goyal and A. Aggarwal, "Composing signatures for misuse intrusion detection system using genetic algorithm in an offline environment," in *Advances in Computing and Information Technology*, ser. Advances in Intelligent Systems and Computing, N. Meghanathan, D. Nagamalai, and N. Chaki, Eds. Springer Berlin Heidelberg, 2012, vol. 176, pp. 151–157.

[16] A. Parashar, P. Saurabh, and B. Verma, "A novel approach for intrusion detection system using artificial immune system," in *Proceedings of All India Seminar on Biomedical Engineering 2012 (AISOBE 2012)*, ser. Lecture Notes in Bioengineering, V. Kumar and M. Bhatele, Eds. Springer India, 2013, pp. 221–229.

[17] T. Sipola, A. Juvonen, and J. Lehtonen, "Anomaly detection from network logs using diffusion maps," in *Engineering Applications of Neural Networks*. Springer, 2011, pp. 172–181.

[18] ——, "Dimensionality reduction framework for detecting anomalies from network logs," *Engineering Intelligent Systems*, 2012.

[19] A. Juvonen and T. Sipola, "Adaptive framework for network traffic classification using dimensionality reduction and clustering," in *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2012 4th International Congress on*. IEEE, 2012, pp. 274–279.

[20] (2013, Aug.) Mit lincoln laboratory: Communication systems and cyber security: Cyber systems and technology: Darpa intrusion detection evaluation. [Online]. Available: http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/

[21] (2013, Aug.) Kdd cup 1999 data. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[22] M. Tavallaee, E. Bagheri, W. Lu, and A.-A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009*, 2009.

[23] J. McHugh, "Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory," *ACM transactions on Information and system Security*, vol. 3, no. 4, pp. 262–294, 2000.

[24] (2013, Aug.) Log files - apache http server. [Online]. Available: http://httpd.apache.org/docs/2.4/logs.html

[25] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: applications to image and text data," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2001, pp. 245–250.

[26] W. Johnson and J. Lindenstrauss, "Extensions of lipschitz mappings into a hilbert space," *Contemporary mathematics*, vol. 26, no. 189-206, pp. 1–1, 1984.

[27] R. Hecht-Nielsen, "Context vectors: general purpose approximate meaning representations self-organized from raw data," *Computational intelligence: Imitating life*, pp. 43–56, 1994.

[28] D. Achlioptas, "Database-friendly random projections," in *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2001, pp. 274–281.

[29] P. Li, T. Hastie, and K. Church, "Very sparse random projections," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 287–296.

[30] P. C. Mahalanobis, "On the generalized distance in statistics," *Proceedings of the National Institute of Sciences (Calcutta)*, vol. 2, pp. 49–55, 1936.

[31] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The mahalanobis distance," *Chemometrics and Intelligent Laboratory Systems*, vol. 50, no. 1, pp. 1–18, 2000.